TP3 : Bibliothèque de cellules pré-caractérisées

Objectifs

Le principal objectif de ce TP3 est d'utiliser le langage VHDL pour écrire une description structurelle hiérarchique multi-niveaux utilisant une bibliothèque de cellules précaractérisées.

Pour cela, nous allons continuer à décomposer les trois blocs **adder**, **mux** et **accu**, définis dans le TP2, en sous blocs, et nous allons finalement décrire chacun des sous-blocs comme une interconnexion de portes de bases, fournies par une bibliothèque de cellules pré-caractérisées (en anglais "standard cells library").

Une cellule pré-caractérisée est une fonction élémentaire pour laquelle on dispose des différentes "vues" permettant son utilisation par des outils CAO:

- vue *physique* : dessin des masques
- vue *logique* : schéma en transistors
- vue *comportementale* : description VHDL

On dit que ces cellules sont pré-caractérisées, car on connait leurs caractéristiques physiques:

- surface occupée
- consommation
- temps de propagation

A) bibliothèque SxLib?

La bibliothèque de cellules utilisée dans ce TP est la bibliothèque SxLib?, développée par le laboratoire LIP6, pour la chaîne de cCAO *ALLIANCE*. La particularité de cette bibliothèque est d'être "portable" : le dessin des masques de fabrication (vue *physique*) utilise une technique de dessin *symbolique*, qui permet d'utiliser cette bibliothèque de cellules pour n'importe quel procédé de fabrication CMOS possédant au moins trois niveaux de métallisation.

Evidemment les caractéristiques physiques (surface occupée, temps de propagation) dépendent du procédé de fabrication. Les cellules que vous utiliserez dans ce TP ont été caractérisées pour un procédé de fabrication CMOS 0.35micron.

La liste des cellules disponibles dans la bibliothèque SxLib? peut être obtenue en consultant la page man :

```
>man sxlib
```

Comme vous pourrez le constater, il existe plusieurs cellules réalisant la même fonction logique. Les deux cellules na2_x1 et na2_x4 réalisent toutes les deux la fonction NAND à 2 entrées, et ne diffèrent entre elles que par leur

puissance électrique: Lacellule na2_x4 est capable de charger une capacité de charge 4 fois plus grande que la cellule na2_x1. Evidemment, plus cellule est puissante, plus la surface de silicium occupée est importante.

B) Schéma des blocs

B1) schéma SxLib? du bloc adder

Un additionneur 4 bits peut être réalisé en interconnectant 4 additionneurs 1 bit suivant le schéma ci-dessous.

Un additionneur 1 bit (encore appelé *Full Adder*) possède 3 entrées a,b,c, et deux sorties s et r. La table de vérité est définie par le tableau ci-dessous. Le bit de "somme" s vaut 1 lorsque le nombre de bits d'entrée égal à 1 est impair. Le bit de "report" est égal à 1 lorsqu'au moins deux bits d'entrée valent 1.

a	b	c	r	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Ceci donne les expressions suivantes :

- s <= a XOR b XOR c
- r <= (a AND b) OR (a AND c) OR (b AND c)

Il existe plusieurs schémas possibles pour réaliser un Full Adder. Nous vous proposons d'utiliser le schéma ci-dessous, qui utilise les deux cellules na2_x1 (NAND 2 entrées), et nxr2_x1 (XOR 2 entrées).

Les étudiants curieux peuvent visualiser le dessin de ces deux cellules en utilisant l'éditeur graphique de la chaîne alliance :

```
> graal -l na2_x1
```

Il faut donc écrire explicitement, en langage VHDL structurel, les deux fichiers adder.vst, et half_adder.vst, correspondant aux deux schémas ci-dessus.

B2) schéma SxLib? du bloc accu

Un registre 4 bits peut être réalisé en interconnectant 4 cellules sff1_x4 suivant le schéma ci-dessous. La cellule sff1_x4 réalise une bascule D à échantillonnage sur front montant. Ceux qui sont curieux peuvent

Il faut donc écrire explicitement, en langage VHDL structurel, les deux fichiers adder.vst, et half_adder.vst, correspondant aux deux schémas ci-dessus.

B3) schéma SxLib? du bloc mux

Il existe plusieurs façons de réaliser un multiplexeur. Un multiplexeur 4 bits peut être réalisé en interconnectant 4 cellules mx2_x2.

C) simulation zero-delay

D) simulation temporelle

Compte-Rendu